2010

How dependencies affect the capability of several feature selection approaches to extract important variables

Qin Yang
University of Windsor

How dependencies affect the capability of several feature selection approaches to extract important variables

by

Qin Yang

A Thesis
Submitted to the Faculty of Graduate Studies
through Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2010

Your file Votre référence
ISBN: 978-0-494-70595-7
Our file Notre référence
ISBN: 978-0-494-70595-7

How dependencies affect the capability of several feature selection approaches to extract of important variables

by

Qin Yang

APPROVED BY:

Dr. Jonathan Wu
Department of Electrical and Computer Engineering

Dr. Dan Wu
School of Computer Science

Dr. Robin Gras, Advisor
School of Computer Science

Dr. Luis Rueda , Chair of Defense
School of Computer Science

August 27, 2010

DECLARATION OF PREVIOUS PUBLICATION

This thesis includes two original papers that have been previously published/submitted for publication in peer reviewed journals, as follows:

Thesis Chapter	Publication title/full citation	Publication status*
Chapter III	*Using feature selection approaches to find the dependent features*	*accepted for publication*
Chapter IV, V	How dependencies affect the capability of several feature selection approaches to extract of the key features	*submitted*

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada

Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

The goal of our research is to find how dependencies affect the capability of several feature selection approaches to extract of the relevant features for a classification purpose. A new method using pre-designed Bayesian Networks is proposed to generate the test datasets with an easy tuning level of complexity. Relief, CFS, NB-GA, NB-BOA, SVM-GA, SVM-BOA and SVM-mBOA these feature selection approaches are used and evaluated. The higher level of dependency among the relevant features can affect the capability to find the relevant features for classification. For Relief, SVM-BOA and SVM-mBOA, if the dependencies among the irrelevant features are altered, the performance changes as well. Relief is an efficient method in normal case except some extreme situations. Moreover, a multi-objective optimization method is used to keep the diversity of the populations in each generation of the BOA search algorithm improving the overall quality of solutions in our experiments.

DEDICATION

This thesis is dedicated to my family and friends.

ACKNOWLEDGEMENTS

My thanks and appreciation to Dr. Robin Gras for persevering with me as my advisor throughout the time it took me to complete this research and write the thesis, and for his guidance on my research and during the course of my graduate study which are the most important experiences in my life.

I am grateful too for the support and advice from Elham Salehi who is a PhD student conducting related topics to mine with my advisor.

My work is supported by the NSERC grant ORGPIN 341854, the CRC grant 950-2-3617 and the CFI grant 203617 and is made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca).

TABLE OF CONTENTS

DECLARATION OF PREVIOUS PUBLICATION iii

ABSTRACT v

DEDICATION vi

ACKNOWLEDGEMENTS vii

LIST OF TABLES x

LIST OF FIGURES xi

CHAPTER

I. INTRODUCTION

Datasets 1
Feature selection 4
Thesis organization 8

II. BACKGROUND AND RELATED WORKS

Feature selection 9
Multi-objective optimization 21

III. EXPERIMENTS BASED ON DATASETS GENERATED BY PARTIALLY RANDOM BAYESIAN NETWORKS

Datasets 24
Experiments 26

IV. EXPERIMENTS BASED ON VARIED DEPENDENCY AMONG THE RELEVANT FEATURES

Datasets 33
Experiments 36

V. EXPERIMENTS BASED ON VARIED DEPENDENCY AMONG THE IRRELEVANT FEATURES

Datasets 41
Experiments 43

VI. CONCLUSION AND FUTURE WORKS

REFERENCES 48

VITA AUCTORIS 53

LIST OF TABLES

TABLE 1. THE CHARACTERISTIC OF DATASET A TO DATASET H 25

TABLE 2. THE EXPERIMENT RESULTS OF NB-BOA AND NB-GA. THE FITNESS FUNCTION PUNISHMENT PARAMETER P=0.0025. 27

TABLE 3. THE EXPERIMENT RESULTS OF NB-BOA AND NB-GA WHERE THE FITNESS FUNCTION PUNISHMENT PARAMETER P=0.005 28

TABLE 4. THE RESULTS OF THE COMPARISON OF USING NB-GA AND NB-BOA. 29

TABLE 5. THE RESULTS OF THE COMPARISON OF CLASSIFICATION ACCURACY OF USING THE SELECTED FEATURES BY NB-GA AND NB-BOA. 29

TABLE 6. THE SUMMARY OF DATASET O, P AND Q 34

TABLE 7. EXPERIMENTS BASED ON DATASETS O,P AND Q 37

TABLE 8. CLASSIFICATION ACCURACY OF THE EXPERIMENS BASED ON DATASETS O, P AND Q 37

TABLE 9. THE SUMMARY OF DATASET R, S AND T 43

TABLE 10 EXPERIMENTS BASED ON DATASETS R, S AND T 44

TABLE 11 CLASSIFICATION ACCURACY OF THE EXPERIMENS BASED ON DATASETS R, S AND T 44

LIST OF FIGURES

FIG. 1. FEATURE SELECTION PROCESS FOR FILTER MODEL 5

FIG. 2. FEATURE SELECTION PROCESS FOR WRAPPER MODEL 5

FIG. 3. STRUCTURE OF NAÏVE BAYES 12

FIG. 4. AN EXAMPLE PROCESS OF SVM 13

FIG. 5. EXAMPLES OF CHOOSING HYPERPLANE FOR SVM 14

FIG. 6. MAXIMUM MARGIN AND SUPPORT VECTOR OF SVM 14

FIG. 7. A SIMPLE EXAMPLE OF BAYESIAN NETWORK 19

FIG. 8. EXAMPLE OF BOA 20

FIG. 9. SOLUTION AND ITS PARETO FRONT 22

FIG. 10. STRUCTURE OF BAYESIAN NETWORK OF CLASS 0 OF DATASET A 25

FIG. 11. STRUCTURE OF BAYESIAN NETWORK OF CLASS 1 OF DATASET A 26

FIG. 12. RELEVANT FEATURES FOR CLASS 0 OF DATASET A 32

FIG. 13. RELEVANT FEATURE FOR CLASS 1 OF DATASET A 32

FIG. 14. BAYESIAN NETWORK STRUCTURE FOR RELEVANT DEPENDENT FEATURES OF CLASS 0 OF DATASET O 35

FIG. 15. BAYESIAN NETWORK STRUCTURE FOR RELEVANT DEPENDENT FEATURES OF CLASS 1 OF DATASET O 35

FIG. 16. THE BAYESIAN NETWORK EXAMPLE FOR DATASET R 42

FIG. 17. THE BAYESIAN NETWORK EXAMPLE FOR DATASET S 42

FIG. 18. THE BAYESIAN NETWORK EXAMPLE FOR DATASET T 42

CHAPTER I

INTRODUCTION

Machine learning, which is now broadly used in many areas, is used to extract some unknown underline knowledge, automatically learn to recognize complex patterns and make intelligent decisions based on datasets. It is a highly challenging problem when the data is high-dimensional and complex, for example, multimedia data, microarrays in genomics and proteomics, and networks in social computing and system biology. Feature selection, also known as variable selection, feature reduction, attributes selection or variable subset selection, is a technique for dimensionality reduction. By removing most irrelevant and redundant features from the dataset, feature selection can helps enhance the capability of prediction, speed up learning process, and helps people better understand about the structure of the data[1-3].

There are many feature selection approaches that have been proposed and broadly used. Some of them are faster; some of them can get higher classification accuracy. We cannot expect that one feature selection approach could be employed for all problems and get good performance. The results may vary depending on the specific properties of the tested datasets. So questions arise here: Which are the suitable tested dataset structure for different feature selection approaches respectively? In our research, we try to find some limitations of several feature selection approaches and focus on how the dependency can affect the capability of these feature selection approaches to extract the relevant features.

Datasets

Dependency, in machine learning dataset, is the mutual dependent relations between features, say, one features value is relying on or being controlled by some other features. Dependencies are directly linked to the complexity of optimization machine learning problems[4]. If all features are independent, we can evaluate them one by one, in contrary, if there are some features mutually dependent, we should consider these dependent features and evaluate them together, the possible search space increased exponentially. Our hypothesis is that more dependencies (more mutually dependent features) and higher level dependencies (several levels of dependencies overlap together, in other words, some features may in several dependencies simultaneously) mean more complexity for feature selection approaches, where the complexity is the amount of time the approach requires to run and get results according to the size of the input to the problem.

To find how the dependency affects the feature selection approach performance to extract the relevant features for classification propose, we should define very clearly the structure of a tested dataset, especially for what the dependent features or the relevant features of the dataset are. A dependent feature is a feature which value is mutually conditionally depended on the value of other features. An independent feature is a feature which value distribution is independent with others. A relevant feature is a feature which is very important to distinguish classes. An irrelevant feature is a feature which is useless to discriminate classes. Our experiments aim to evaluate the capability of several feature selection approaches to extract the relevant features based on the datasets which include two class data for classification. As the most multi-classes classification problems can be transferred to several 2-classification problems and to focus on the effect of the main

factor (dependency), we limit our experiments, without loss of generality, to the classification problems with two classes.

At present time, feature selection experiments are often based on real datasets (e.g. UCI Machine Learning Repository [5]) or some simple artificial datasets. Both of them have some problems. For the artificial datasets, they are usually very simple; for example, the tested datasets of XOR problem or MONK's Problems [6]. They have few features with few dependencies and not sufficient to depict a large and complicated problem we want to test. Moreover, for feature selection experiments based on real datasets, we do not know what the exact dependencies among the dataset are. We also do not know the exact relevant features to distinguish the classes (we could obtain different selected features by different feature selection approaches with similar classification accuracy and we do not know which one is the final answer). Therefore, these two kinds of dataset are not suitable for our experiments. We want for our dataset: flexibility, tuneable complexity, and possibility to distinguish relevant or irrelevant features, dependent and independent features... As we cannot find in the literature any dataset that corresponds to the criterion we have defined, we need to conceive a framework to generate datasets that can be used to precisely test the efficiency of the different feature selection methods.

To solve this problem, we use Bayesian Networks to generate the test datasets. Bayesian Network, which is a probabilistic graphical model, is usually used to represent the conditional probabilities of different situations. There exists several algorithmic method to learn a Bayesian network from training datasets [7][8]. Compare with other probabilistic models, one advantage of Bayesian Network is that it can represent complicated dependencies among the features. We use Bayesian Network in an inversely

way from learning. Saheli et al. used the Bayesian Networks to generate the tested datasets for comparison local search heuristics algorithms for Bayesian network structure learning and for detecting multiple dependencies [9, 10]. The new idea in our experiments is that we first pre-design a Bayesian network, and we set every properties needed to simulate different types of the Bayesian Network, and then we use this Bayesian Network to generate our test datasets. These datasets do not have exactly the same properties as real datasets, but they still are very close to what we can expect from real datasets.

In our experiments, each dataset include two class datasets which are generated by a pair of pre-designed Bayesian Networks, one for class0 and one for class1. The Bayesian networks are the probabilistic models we use to represent the joint probabilistic distribution of each class in the dataset. The nodes' position, links (edges) between nodes and the values of the conditional probability distribution tables of the Bayesian network are easily adjustable to simulate different test situations. This method can generate very complicated test datasets for which we exactly know and be able to change their structure, dependencies, distribution, and relevant and irrelevant features for our different experiment purposes.

Feature selection

Feature selection problem has been defined as : the identification of a minimal subset of features that are relevant to the target concept [11]. These features are necessary and sufficient to describe the target concept. The feature selection approach is a method to find these relevant features in the candidate feature sets.

Most typical feature subset selection approaches have two major parts: 1. a generation procedure to generate the next candidate subset; 2. an evaluation function to evaluate the feature subset. Based on the evaluation criterion, feature selection methods can be divided into filter model and wrapper model [12]. Filter Model: select good features based on the certain data intrinsic properties [11, 13, 14]. These approaches measure the relevance of a feature to the target class. It can be further grouped into distance, consistency, and correlation measures [11]. Relief [15] and the Correlation-based Feature Selection (CFS) [14] are the two typical filter model feature selection approaches. Relief uses a distance measure to distinguish different features. CFS measures the dependencies or correlation to predict the features that belong to one class or another. Fig. 1 and Fig. 2 are the summarization of the filter and the wrapper model feature selection.

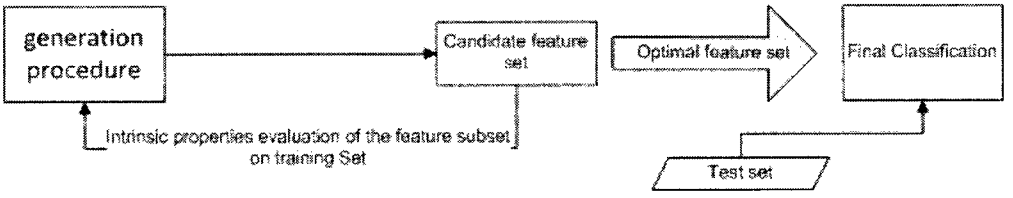

Fig. 1. Feature selection process for filter Model

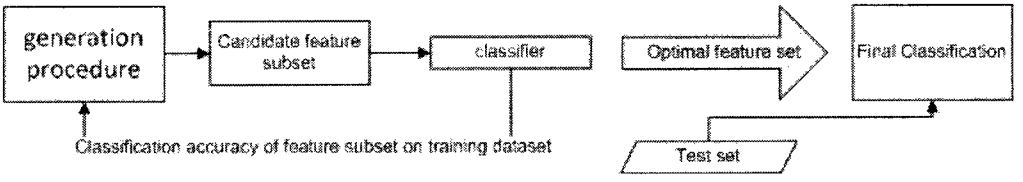

Fig. 2. Feature selection process for wrapper Model

Wrapper Model feature selection uses classification accuracy as an evaluation criterion for the candidate feature subsets. The feature set with the highest classification accuracy is considered as the best feature set. Normally, it has higher classification

accuracy and a larger computation workload than the filter model. There are lots of classifiers that can be chosen, e.g. Decision Tree [16], Knn classifier [17], Naive Bayes [18] etc.

The generation procedure of feature selection can be considered as a search problem, thus each state in the search space represents a subset of the possible features of the task. Roughly speaking, these search algorithms can be categorized into exact algorithms (e.g. depth-first [19], breadth-first and Branch & Bound search [20] etc.) and heuristic algorithms (e.g. SFS, SBS [11], Genetic Algorithms [21], Simulated Annealing [22] etc.). Obviously, for a large feature set, exhaustive evaluation of possible feature subsets is usually unfeasible because of the exponential computational time requirements.

Genetic Algorithms (GAs) are randomized, evolutionary and population-based search algorithms that are proposed to solve these problems. They are inspired by biological evolution: reproduction, mutation and selection. They use fitness function to evaluate candidate solutions. The better instances have more chances to "live" in the next generation. Evolution then takes place after the repeated application of the above operators. GAs have been broadly used in many different areas. One drawback of GAs is that problem independent recombination operators of GAs may break some good building blocks (which are partial solutions of the problem, formed by groups of related features), and cause to produce a convergence to a local optimal or delay the discovery of the global optimal. That can be also understood as the fact that GAs does not use the information about the dependencies among the related features. The Estimate of Distribution algorithms (EDAs) [23, 24], which are another kind of evolutionary and population-based search algorithms, have been proposed to solve these problems. There

are no mutation and crossover operators and the new population is sampled from a probabilistic distribution which is estimated from the selected solutions. EDAs perform efficient mixing of key substructures or building-blocks (BBs). They also provide additional information about the problem being solved. The probabilistic model of the population that represents the dependencies among relevant features is an important source of information that can be exploited and used to enhance the performance of EDAs. It can also assist the user within a better interpretation and understanding of the underlying structure of the problem.

In our research, the capability of several feature selection approaches to extract the relevant features on different artificial datasets which have varied dependency levels and structures have been tested. For the filter model, the feature selection approaches are Relief and CFS. Relief is a classic filter model feature selection approach which evaluates distance between features and target class and ranks them, CFS is another one based on computation of correlation values. For the wrapper model, a classical genetic algorithm (GA) and Bayesian Optimization Algorithms (BOA) [25] have been used as the search algorithm for feature subset generation procedures (BOA is one of EDAs). GAs or EDAs select a group of solutions which can be used to extract some properties of search space. These properties are directly linked to the dependency. We expect that the properties which will be extracted during the exploration will be useful to localize the relevant features. Naive Bayes (NB), a simple but robust and fast classification classifier, and Support Vector machine (SVM) which is popular and able to handle dependencies among features, are the classifiers for the wrapper model feature selection in our experiments. These feature selection approaches are NB-BOA, NB-GA, SVM-BOA and SVM-GA.

SVM-mBOA is a feature selection approach which combines SVM-BOA and a multi-objective optimization method together. In this case, the multi-objective optimization method not only help us choose a trade-off solution from two conflict objectives (high classification accuracy and less selected features) but also be used to improve the search capability of BOA.

Thesis organization

Chapter I is the introduction of thesis. Chapter II is background and some related works. Our experiments are divided into three parts and is covered in chapters III, IV and V: one to evaluate several selected feature selection approaches to extract the relevant features for classification problems with random dependencies among the dependent features; one to find how the varied dependency among the relevant features can affect the capability of the feature selection approaches to extract the relevant features and one to show how the different dependencies among the irrelevant features can affect the results of extracting those relevant features with these feature selection approaches.

We use capital letters like X, Y, Z... for the instances in datasets. Each instance composed of n features e.g. $(x_1, x_2 \ldots x_n)$. Lower case letters a, b, c ... represent the training datasets.

CHAPTER II

BACKGROUND AND RELATED WORKS

In our experiments, we use Bayesian Network, to generate our tested datasets and Naïve Bayes and SVM as the typical and popular classifiers have been chosen in our experiments. GA and BOA, which are good for finding dependency among the features to help locate the relevant features, are the search algorithms of our wrapper approach. Multi-objective optimization can improve the search ability of BOA. We therefore add a method including multi-objective optimization in our set of compared approaches. Relief and CFS are the very classic filter model feature selection methods which we are interested to evaluate. These methods and algorithms which had been used in our experiments are briefly introduced in this chapter. We first begin with feature selection.

Feature selection

The Filter model feature selection

The filter model uses some intrinsic properties such as distance, consistency, and correlation of the data to select the optimal feature set. Relief measures the distance and Correlation-based Feature Selection (CFS) calculates the correlation of the feature sets. Both are typical filter approaches which we used in our experiments.

Relief

Relief [15] is a feature weight based algorithm. Relief detects those features which are statistically relevant to the target concept. Differences of feature values between two instances X and Y are defined by the following function diff.

When x_k *and* y_k are nominal,

$$diff(x_k, y_k) = \begin{cases} 0 \; if \; x_k \; and \; y_k \; are \; the \; same \\ 1 \; if \; x_k \; and \; y_k \; are \; different \end{cases}$$

When x_k *and* y_k are numerical,

$$diff(x_k, y_k) = (x_k - y_k)/nu_k$$

nu_k is a normalization unit used to normalize the values of diff into the interval [0, 1]. Relief randomly picks m instances from the dataset and calculates each solution's Near-hit instance (the closest instance according to Euclidean Distance in the same class), and Near-miss instance (the closest instance according to Euclidean Distance in the opposite class). It updates the feature weight vector W for all m samples to determine the average feature relevance weight vector (of all the features to the target concept). Finally, Relief selects those features whose average weights ('relevance level') are above a given threshold τ. Here is the pseudo Code of Relief

1. initial the weight vector $W := 0$;
2. for i:=1 to m do begin
3. randomly select an instance $X(x_1, x_2 \ldots x_n)$;
4. find Near-hit $Y(y_1, y_2 \ldots y_n)$ and Near-miss instance $Z(z_1, z_2 \ldots z_n)$;
5. for k:=1 to # all_features do
6. $W[k] := W[k] - diff(x_k, y_k)^2 + diff(x_k, z_k)^2$
7. end for
8. for k:=1 to # all_features do
9. if $W[k] > \tau$ then $feature_k$ is a relevant feature
10. Else $feature_k$ is a irrelevant feature;

Correlation-based Feature Selection (CFS)

The key point of the CFS [14] algorithm is a heuristic for evaluation of the worth or merit of subset features. The equation formalizing the heuristic is:

$$Merit_s = \frac{k\overline{r}_{cf}}{\sqrt{k + k(k-1)\overline{r}_{ff}}}$$

Where $Merit_s$ is the heuristic "merit" of a feature subset s containing k features, $\overline{r_{cf}}$ is the average feature-class correlation, and, $\overline{r_{ff}}$ is the average feature-feature inter-correlation.

In order to apply this equation to estimate the merit of feature subsets, CFS uses symmetrical uncertainty (SU, a modified information gaining measure) [26] to estimate the degrees of correlation between discrete features.

$$SU = 2.0 \times \left[\frac{H(x_i) + H(x_j) - H(x_i, x_j)}{H(x_i) + H(x_j)} \right]$$

The Wrapper Model Feature Selection

The wrapper model feature selection approaches evaluate the candidate feature subsets by their classification accuracy. There are two important components for this approach. One is the classifier and the other is the feature subset generation procedure.

Naive Bayes classifier

Naive Bayes (Fig. 3) classifier is based on a very simple Bayesian Network in which all the features depend directly on the class and are statistically independent of each other. For example, in figure 4, C is the class label and X1…X5 are the features.

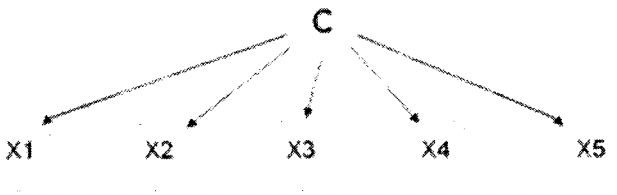

Fig. 3. Structure of Naïve Bayes

It follows usual steps of a Bayesian-Network based classifier. The Bayesian network is learned from the learning datasets, first learning the structure (here already given in Naïve Bayes), and then learning the conditional probabilities tables. Then to classify an instance, the algorithm evaluates the probabilities for this instance to belong to one class or another, conditionally to the features of this instance.

$$p(C \mid X) = p(C) \bullet \Pi_{i=1}^{n} p(x_i \mid C)$$

C is a class label, X is an instance which contain n features($x_1, x_2 \ldots x_n$). This very simple relation between the features allows faster calculations and in most cases, it also can get good classification accuracy even though sometimes the features are not total independent.

Support Vector Machine (SVM) classifier

Support vector machine (SVM), which are often used for classification, regression, or other tasks, is a supervised leaning method. It maps the samples from an original lower-dimensional space in which the samples are hard to be discriminated to a higher-dimensional new space. In this new space, SVMs finds the separating hyper plane with the largest margin; so that the samples can be easily separated by a linear or non-linear discriminant function. Samples on the margin are called the support vectors. Kernel

functions are used to map samples from original space to a higher-dimensional new space. : Fig. 4. is an example.

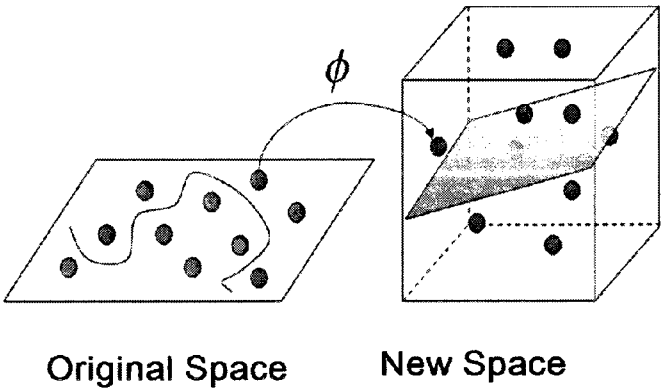

Fig. 4. An example process of SVM

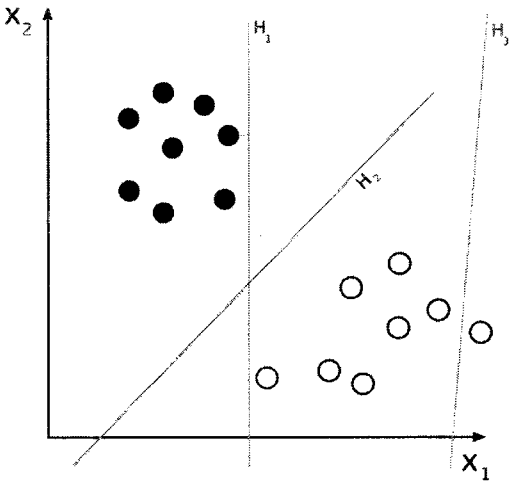

Fig. 5. Examples of choosing hyperplane for SVM1

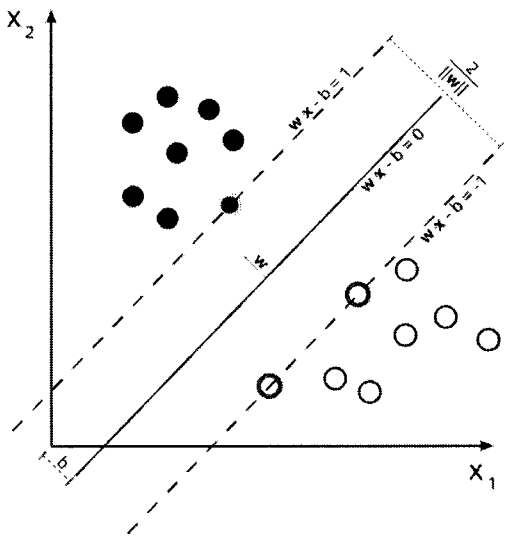

Fig. 6. Maximum margin and Support Vector of SVM2

In

Fig. 5 H3 (green) doesn't separate the 2 classes. H1 (blue) does, with a small margin, and H2 (red) with the maximum margin. Fig. 6 illustrates the Maximum-margin hyper plane and margins for a SVMs trained with samples from two classes. Below are some common kernels functions:

Polynomial (Homogeneous):

---

[1] Wikipedia, http://en.wikipedia.org/wiki/Support_vector_machine

[2] Wikipedia, http://en.wikipedia.org/wiki/Support_vector_machine

$$k(x_i, x_j) = (x_i \bullet x_j)^d$$

Polynomial (inhomogeneous):

$$k(x_i, x_j) = (x_i \bullet x_j + 1)^d$$

Radial Basis Function:

$$k(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2), \text{for } \gamma > 0$$

Gaussian Radial basis function:

$$k(x_i, x_j) = \exp(-\frac{\parallel x_i - x_j \parallel}{2\sigma^2})$$

Hyperbolic tangent:

$$k(x_i, x_j) = \tanh(\kappa x_i \bullet x_j + c), \text{for some (not every) } \kappa > 0 \text{ and } c < 0$$

Feature subset generation procedure

For wrapper model feature selection, a feature subset generation procedure is used to generate the new candidate feature sets. How to find possible feature subsets is a combinatorial optimization problem, which is to find the least costly solution into a solution space in which each solution is associated with a numerical cost. For our problem a solution is a subset of the feature of the problem. As 2^n subsets can be built with n different feature, the combinatorial optimization problem we face on has an exponentially large solution space with the number of features. In our tests, each dataset has 10,000 instances, and each instance has 100 features (except dataset r, which has 25 features). There are therefore a maximum of 2^{100} possible solutions that should be evaluated by the generation procedure. The exact search algorithms, because the size of the search space is exponential with the number of features, are not feasible in our experiments. Local search algorithms such as Tabu search, simulated annealing and hill

climbing use only a single solution at a time which is modified to explore the search space guided by an objective function. On the contrary, GAs or EDAs are population based search algorithms which mean that they use a sample of the search space instead of a single solution. This sample can be used to extract some properties about the search space which can be useful to improve exploration performance. These properties are directly linked to the dependent features. We expect that the properties which will be extracted during the exploration will be useful to localize the relevant features.

EDAs, and a typical sample of it: BOA, which uses Bayesian Network as a probabilistic model, do not use mutation and crossover operators. Contrary to GAs, these algorithms explicitly search for the dependencies among features, and use the population sample as a training set for building a probabilistic model. The new population is sampled from the probabilistic distribution by the probabilistic model. BOA uses a Bayesian Network as the probabilistic model to estimate the joint distribution of promising solutions, and then samples the new solutions (the new population) from the joint probability distribution encode by the Bayesian Network.

Genetic Algorithms (GAs)

Genetic Algorithms (GAs) are randomized, evolutionary and population-based search algorithms. They are inspired by biological evolution, and they use mutation, recombination, and selection operators for evolution, and use a fitness function to evaluate candidate solutions. Below is the pseudo code of GAs.

1. Randomly generate initial population of candidate solutions
2. Evaluate the fitness of each solution in the population
3. Select the best-fit solutions for reproduction;

4. Breed new solutions through crossover and mutation operations to give birth to offspring

5. Evaluate the solution fitness of new solutions

6. If the termination criteria are not met, go to (2)

Estimate of Distribution algorithms (EDAs)

Estimate of Distribution algorithms (EDAs) are evolutionary and population-based search algorithms which do not use mutation and crossover operators. Contrary to GAs, these algorithms explicitly search for the dependencies among features, and use the population sample as a training set for building a probabilistic model. The new population is sampled from the probabilistic distribution by the probabilistic model.

The pseudo code of EDAs is:

1. Randomly generate initial population

2. Evaluate the fitness of each solution in the population

3. Select the best-fit solutions for reproduction, and estimate the joint probability distribution among the selected solutions.

4. Sample the solutions (the new population) from the joint probability distribution.

5. Evaluate the solution fitness of the new solutions

6. If the termination criteria are not met, go to (2)

Bayesian Optimization Algorithms (BOA)

Bayesian Network is a probabilistic graphical model. It is represented by a directed and acyclic graph, composed of a set of nodes and directed edges. In Bayesian networks, the nodes represent random variables in the Bayesian sense and the edges

represent conditional dependencies. Nodes which are not connected represent variables which are conditionally independent of each other. There is a finite set of mutually exclusive states for each node. Each node is associated with a conditional probability distribution table (CPT) that takes as input a particular set of values for the incoming node's variables (its parent variables) and gives the probability of the variable represented by the node. For each variable X_i with parents Y_1, Y_2 to Y_k $k \leq n$, there is an attached probability table $p(X_i|Y_1, Y_2 \ldots, Y_k)$.

The joint probability distribution for X is:

$$p(X) = \prod_{i=1}^{n} p(x_i|\pi_{xi})$$

Where π_{xi} is the parents of X_i (the set of nodes from which have an edge toward X_i) in the Bayesian Network.

Bayesian network has been used to represent the probabilistic relationships between results and reasons. Given results, the network can be used to compute the probabilities of the presence of various reasons. Fig. 7 is an example of Bayesian Networks, which can usually be learned from statistical data. However, in our experiments, we pre-design a Bayesian Network like this one, we can therefore simulate any, even some extremely, situations only by change some properties of network, and then use it to generate our tested datasets.

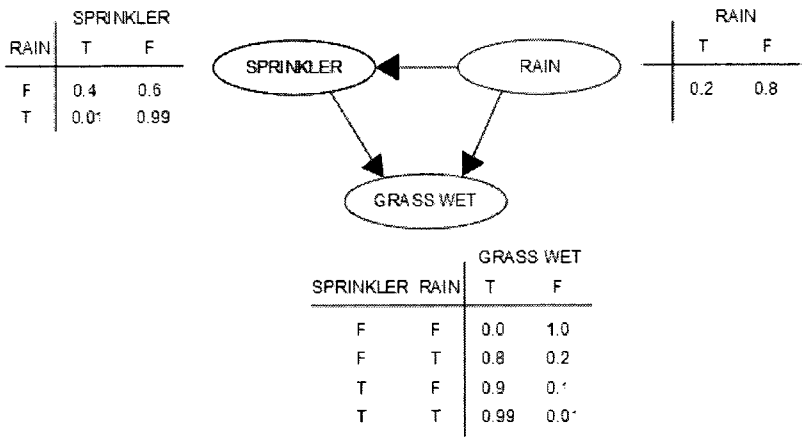

Fig. 7. A Simple Example of Bayesian Network

Bayesian Optimization Algorithm (BOA) is one EDAs. It uses a Bayesian Network as a probabilistic model to estimate the joint distribution of promising solutions. BOA needs to construct the Bayesian network by using a chosen metric and constraints using the selected solutions, and then samples the new solutions (the new population) from the joint probability distribution encode by the Bayesian Network. Fig. 8 is an example to illuminate the process of BOA.

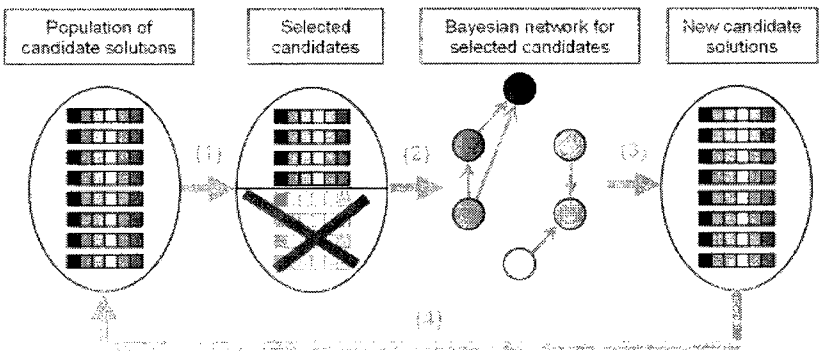

Fig. 8. Example of BOA3

The pseudo code of BOAs is:

1. Randomly generate initial population

2. Evaluate the fitness of each solution in the population

3. Select the best-fit solutions for reproduction, and estimate the joint probability distribution among the selected solutions, generate a Bayesian Network to represent the probabilistic distribution model of optimistic solutions.

4. Sample the solutions (the new population) from the joint probability distribution model.

---

[33] Martin Pelikan http://www.cs.umsl.edu/~pelikan/boa.html

5. Evaluate the solution fitness of the new solutions

6. If the termination criteria are not met, go to (2)

Multi-objective optimization

Finding an optimal solution or optimal solutions in problems which have two or more objectives are called Multi-objective optimization. The difficulty comes from the fact that sometimes the objectives are in conflict with each others. For example, in our experiments, we try to select fewer features with higher classification accuracy. However, for most classifiers, using more features the classification might be more precise. As a result, finding a final solution often should be a trade-off process. Multi-objective problems are sometimes converted into single-objective problems by assigning weights to the different objectives, and calculating a single fitness value (for example the sum or the product of the weighted objectives). But how to set the weight is a subjective process. For most cases, there is no a priori knowledge of the importance of different objectives, which mean this work is very hard or impossible.

Multi-objective optimization is an optimization approach using Pareto compliant ranking of solutions to solve this problem. It evaluates solutions for all objectives and finds a Pareto front. We can easily explain it by the dominance concept: A solution C dominates a candidate solution D when C is better than D on at least one objective and not worse on others. For example, using only cost and performance to describe a car, car A dominates car B mean that car A has better performance than B and at same time it needs equal or less cost; on the contrary, if car A has better performance but also is more expensive than car B, in this situation, car B is not dominated by car A. The Pareto front is a subset of all solutions that are not dominated by any other solutions. Fig. 9 is an

example. For solutions in rank one (Pareto front, square point), there are no other solutions dominate them. For solution of Rank two, there are no others dominating them except solutions of rank one, and so on.

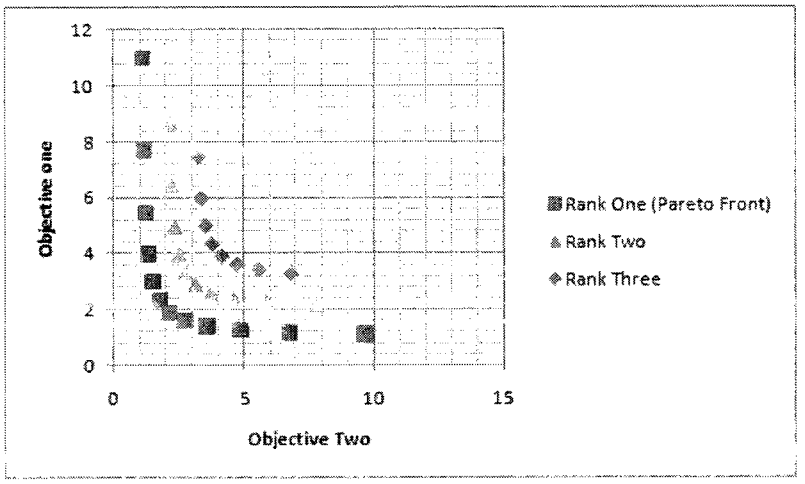

Fig. 9. Solution and its Pareto Front

We give here the pseudo codes of procedures of Multi-objective optimization in our experiments:

Dominate (A, B); // A, B are solutions of population

1. If (accuracy (A) > accuracy (B)) and (featureNumber (A) <= featureNumber (B))
2. then return True;
3. If (featureNumber (A) < featureNumber (B)) and (accuracy (A) >= accuracy (B))
4. then return True;
5. Return False;

Non-dominated sorting:

```
1.  rank = 1
2.  Non-dominated_sorting (p, rank) // p is the set needing to be ranked
3.      If  (p is null) then return;
4.      For each solution (A) in p {
5.          If (no one else dominates A in p) then (A.rank=rank);
6.          p=p-{A};}
7.       Non-dominated_sorting (p, rank+1);
8.   Return;
```

Compare (A, B)

```
1.  If (rank (A) < rank (B)) then A is better than B;
2.  If (rank (A) < rank (B)) then B is better than A;
3.  If (rank (A) = rank (B))
4.  then If (accuracy (A) > accuracy (B)) then A is better than B;
5.      If (accuracy (A) < accuracy (B)) then B is better than A;
6.      If (accuracy (A) = accuracy (B)) then A is better than B;
```

CHAPTER III

EXPERIMENTS BASED ON DATASETS GENERATED BY PARTIALLY RANDOM BAYESIAN NETWORKS

Datasets

In this part, we evaluated several selected feature selection approaches to extract the relevant features for classification problems. There are eight test datasets, each dataset include 2 class data, and 5000 instances for each, each instance having 100 features, 25 dependent features and 75 independent features. Every feature have three possible values : Value1, Value2 and Value3. We use a different Bayesian Network for each class to represent the distribution of a subset of the 100 features. The 75 independent features have the same distribution for both class and this distribution is independent from all the other features. Which mean that only of the 25 features that are dependent can discriminate the two classes. According to our definition before, they also are the relevant features and this is also why our wrapper feature selection approach will try to select them. In the experiments, eight pairs of Bayesian Networks are used to randomly generate the corresponding datasets "a" to "h" under some restrictions. The restrictions are: "the maximum degree of dependency (maximum size of any parent set)", " the number of dependencies (edges in the Bayesian network)", "the 75 independent features have same distribution" and "the position of dependent features in the network". The dependencies among the dependent features and their distributions are generated randomly.

Dataset	Distribution of I	K	A

a	Random	5	40
b	(80,10,10)	5	40
c	(30,35,35)	5	40
d	Random	2	40
e	(80,10,10)	2	40
f	Random	5	70
g	Random	10	70
h	Random	10	120

Table 1. The characteristic of dataset a to dataset h

In **Table 1**, which is the summary of datasets "a" to "h", I is the distribution of independent features, "Random" mean the feature's initial distribution is randomly chosen, say "(23,16,61)" or "(55,27,18)", and these distributions varies for one feature from the other. "(80, 10, 10)" mean the value of this feature is 80% for "Value1", 10% for "Value2" and 10% for "Value3", and so on; both classes are same in a dataset. K is the maximum degree of dependency (maximum size of any parent set) and A is the number of dependencies (edges in the bayesian network). **Fig. 10** and **Fig. 11** are an example of the different classes of the Bayesian Network structure.

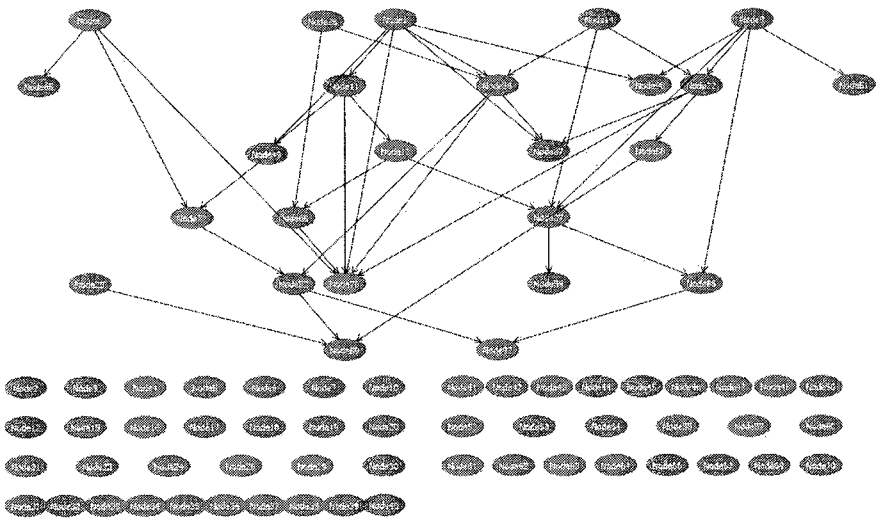

Fig. 10. Structure of Bayesian Network of class 0 of dataset a

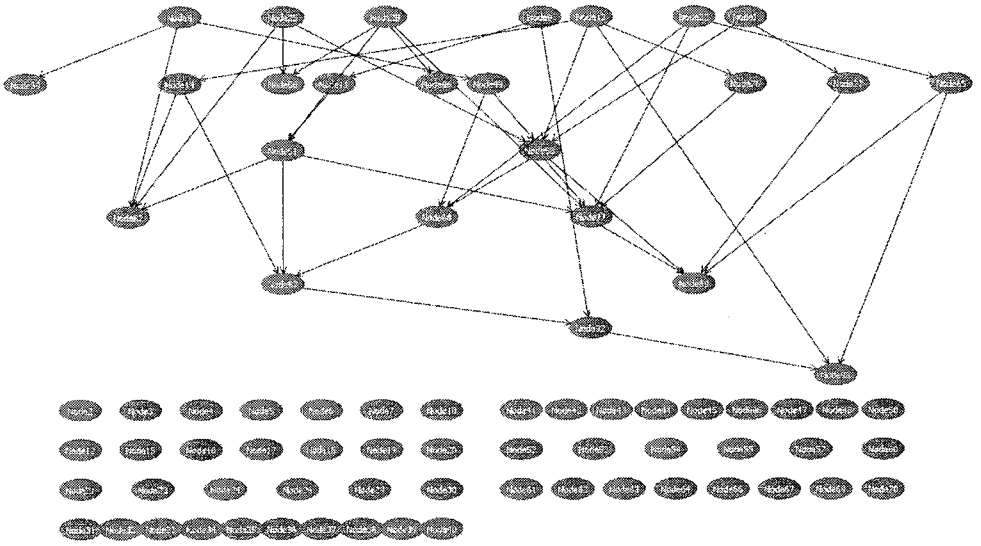

Fig. 11. Structure of Bayesian Network of class 1 of dataset a

Experiments

We tested the capability of different feature subset selection approaches on artificial dataset "a" to "h", which have different degrees of difficulty depending on the complexity of the dependency network model and on the different kinds of independent distribution. For the filter model, we use Relief, and CFS. The threshold τ of Relief is set to 0, which mean all the features which have any connection with the target class will be selected, even though the connection is a tiny one. The programs we used are implemented by Weka [27]. The results are presented in Table 4. For the Wrapper model, we used the Naïve Bayes as the classifier with GAs and BOA search algorithm respectively (NB-GA, NB-BOA). BOA is coded by Pelikan [28]. The fitness function we used is:

$$fitness = Acc./n^p$$

$Acc.$ is the accuracy of the Naïve Bayes classifier, n is the number of selected features and p is an adjustment coefficient. The goal is to find small feature sets with

high classification accuracy. This is also a way to solve the multi-objetive problem. It switches the two objectives (less feature number and higher classification accuracy) into one objective (higher fitness value). We concentrated on finding small feature set with high classification accuracy. Different p values can bring very different results (Table 2, Table 3, Table 4 and Table 5, Experiments are based on an Acer Desktop, Intel Core 2 Quad CPU, 4G memory, Operating system is Ubuntu 9.04. These results are the average value based on 12 times run with 3 fold cross validation).

NB-BOA(P=0.0025)						NB-GA(P=0.0025)				
Dataset	Gens.	Time(s)	Fitness Value	Relevant features	selected features	Gens.	Time(s)	Fitness Value	Relevant features	selected features
a	31.17	272.4	0.9676	19	25.3	41.67	320.58	0.9677	18.8	24.8
b	32.92	286.4	0.9678	19	25.7	40.5	321.33	0.9679	19	24.9
c	31.58	279.1	0.9678	19.3	27.3	37.83	298.42	0.9679	19	27.7
d	41.75	382.4	0.8841	19.1	45.1	52.58	443.83	0.8843	19.9	47.4
e	37.33	336.7	0.8830	21.5	36	53.5	438.83	0.8832	20.9	37.1
f	34.5	289.1	0.9495	16.2	25.4	44.08	333.33	0.9495	16.3	23.8
g	39	348.8	0.9152	16.6	39	49.83	409.92	0.9153	16.8	40.3
h	41.83	388.3	0.8062	18.1	49.3	47.75	404.58	0.8065	17.8	47.1

Table 2. The experiment results of NB-BOA and NB-GA. The fitness function punishment parameter p=0.0025.

In Table 2, Table 3, Table 4 and Table 5 "Gens" is the number of generations, and "time(s)" is the total running time in seconds needed for the NB-BOA and NB-GA to converge. "Relevant dependent features" is the number of real relevant dependent features discovered by our approaches. "Select features" is the total number of selected features by the feature selection approaches.

NB-BOA(P=0.005)					NB-GA(P=0.005)			
Dataset	Gens.	Fitness Value	Relevant features	selected features	Gens.	Fitness Value	Relevant features	selected features
a	25.33	0.9605	18	18.5	28	0.9605	18.8	18.8
b	24.75	0.9605	18.5	18.9	28.75	0.9605	18.7	18.7
c	25.58	0.9604	19	20.3	28.17	0.9605	19	20

d	35	0.8761	19.1	32.8	45.67	0.8762	19.8	33.3
e	32.17	0.8756	20.5	27.9	38.5	0.8758	21.1	27.2
f	25.83	0.9428	14.2	14.7	31.67	0.9428	13.8	14.4
g	31.58	0.9076	16	22.6	38.25	0.9078	16	21.2
h	39.58	0.7984	17.3	41.3	48.42	0.7986	17.2	41.5

Table 3. The experiment results of NB-BOA and NB-GA where the fitness function punishment parameter p=0.005.

Datasets "a" to "h" are generated by partially random Bayesian Networks according to the method we mentioned before. The results of the classification accuracy for selected features by using different feature selection approaches are shown in Table 5. The classifier is Naïve Bayes with 3-fold cross validation. NB-BOA and NB-GA can get better classification performance than CFS and Relief when they have a similar number of selected features. But we just focus on finding the influence of the relevant features among the data. The main measure we used to evaluate each approach was the total number of real relevant features, or relevant key features, and the selected features.

The value in the Table 2 and Table 3 is an average value over 12 runs. According to Table 2 and Table 3, the NB-BOA and NB-GA got very similar results: similar fitness value, similar number of founded dependent features, and similar number of selected features. NB-BOA needs less generation for convergence. But considering that it needs more time to construct the Bayesian networks, the overall running time is not much less than NB-GA.

			NB-BOA			NB-GA		
Dataset	CFS	Relief	(p=0.0025)	(p=0.005)	(p=0.01)	(p=0.0025)	(p=0.005)	(p=0.01)
a	14/15	25/28	19/23	18/19	15/15	19/25	19/19	13/13
b	14/14	25/28	19/26	19/19	15/15	19/25	19/19	14/14
c	14/14	25/26	19/27	19/20	15/15	20/28	19/20	14/14
d	11/14	25/26	19/45	19/32	19/21	20/47	20/33	19/21
e	11/13	25/28	22/36	21/28	19/21	22/38	21/27	18/19

f	10/11	25/26	16/25	14/15	10/11	16/24	13/14	9/9
g	9/11	25/26	17/39	16/23	16/17	17/40	16/21	16/17
h	8/10	19/22	18/49	17/41	13/17	17/49	17/42	13/15

Table 4. The results of the comparison of using NB-GA and NB-BOA.

In Table 4, Where 14/15 mean " 14 relevant features selected among / 15 selected features". The threshold τ of Relief is 0.

			NB-BOA			NB-GA		
Dataset	CFS	Relief	(p=0.0025)	(p=0.005)	(p=0.01)	(p=0.0025)	(p=0.005)	(p=0.01)
a	0.9714	0.9729	0.9754	0.9746	0.9739	0.9755	0.9747	0.9731
b	0.9710	0.9727	0.9756	0.9747	0.9738	0.9757	0.9746	0.9733
c	0.9710	0.9728	0.9758	0.9750	0.9739	0.9759	0.9750	0.9734
d	0.8735	0.8840	0.8926	0.8915	0.8890	0.8928	0.8917	0.8893
e	0.8731	0.8837	0.8909	0.8903	0.8888	0.8912	0.8904	0.8886
f	0.9517	0.9483	0.9572	0.9555	0.9537	0.9571	0.9554	0.9524
g	0.9053	0.9144	0.9236	0.9219	0.9204	0.9238	0.9218	0.9203
h	0.7296	0.8040	0.8141	0.8134	0.8080	0.8143	0.8136	0.8074

Table 5. The results of the comparison of classification accuracy of using the selected features by NB-GA and NB-BOA.

In the experiments based on the dataset from "a" to "h", we found that even though the dataset has become more complicated, Relief always has a good performance, whereas, the performances of CFS, NB-GA and NB-BOA become worse (Table 4). But the performance of relief also decreases significantly for a very complicated network.

We had expected that, because Relief uses a statistical method, only corresponding to the feature individually, it would obtain poor results for these problems as the selected features are all mutually dependent. Classic genetic algorithm uses problem independent recombination operators which do not use the information about the dependencies among the decision features; this is contrary to BOA (or other EDAs). CFS computes the correlation (dependency) between the feature subsets and class by

measuring the information gained. In theory, we anticipated that the feature selection approach using BOA should be good at finding the relevant dependent features compared with those approaches using the genetic algorithms, Relief and CFS. However, in the experiments based on datasets a to h, we found that even though the dataset had become more complicated, Relief always had a good performance, whereas the performances of CFS, NB-GA and NB-BOA become worse (Table 4). BOA only has similar performances with GAs to find the relevant key features in our experiments. Also, we have found that increasing the p value will reduce the selected features and slightly decrease the classification accuracy in some domains. For example, when p is equal to 0.0025, based on dataset a, NB-BOA on average found 19 relevant features over 25 totally selected features, the classification accuracy is 0.9754, when p is 0.005, the result is 18/19 and accuracy is 0.9746. We are using p to adjust the influence of the number of selected features in our fitness function. However, how to set p is a trade-off process. It is a difficult problem that needs lots of experiments and prior knowledge, In Chapter IV and V, we try to use multi-objective method to solve this problem

Another interseting thing, we found that those relevant features which were not selected by NB-BOA or NB-GA had a big chance to be a leaf node or to had a linked arc to a unselected node in the Bayesian Networks (Fig. 12 and Fig. 13, the nodes with a red circle mean unselected features). According to the previous experiments, we can understand that there should be two reasons: Either they are worthless for classification or they have very complicated dependencies that the classifier cannot determine. But we did not do some further analysis and experiments, The exact reason may be work for future experiments.

Several questions arise here: Why do approaches like BOA, which can handle dependency between features, have a worse performance than relief? It may be due to the fact that the classifier we used, Naïve Bayes, can not represent dependencies between features (we also tested with the Decision Tree classifier which can represent some kind of dependencies, but the results we obtained are worse than the results we obtained using Naive Bayes classifier). It may be that the "strength" of dependencies (or link strengths [29], which measures the level of dependency) between these relevant dependent features are too weak. BOA could consider some weakly dependent relevant features as independent features and ignores them. On the other hand, the fact that our datasets do not include redundant features is an advantage for Relief, as one of the known problems of Relief is that it has difficulties to filter redundant features. Our dataset do not present these difficulties for the Relief approaches. To answer these questions, we decide to do more detail experiments by using some specific datasets. We will introduce them on next chapter.

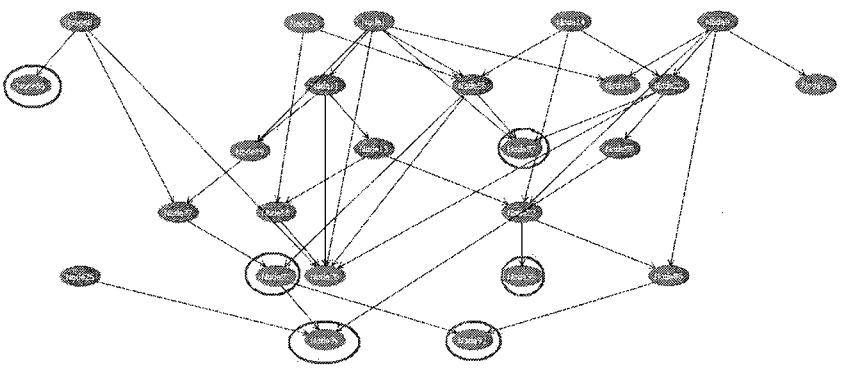

Fig. 12. Relevant features for class 0 of dataset a

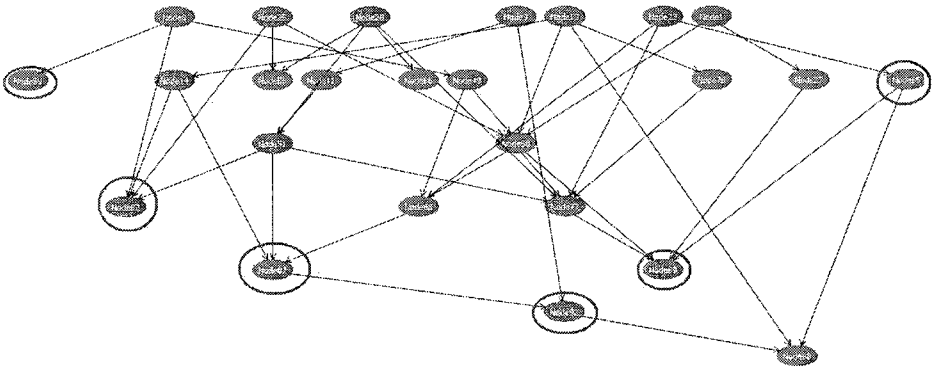

Fig. 13. Relevant feature for class 1 of dataset a

CHAPTER IV

EXPERIMENTS BASED ON VARIED DEPENDENCY AMONG THE RELEVANT FEATURES

Datasets

In this part, we want to know how the varying the dependencies among the relevant features can affect the performance of the selected feature selection approaches to extract the relevant features for classification problems. Three pairs of Bayesian Networks are used to generate the corresponding datasets o, p and q, which have two, three and four dependencies among the relevant features. Similarly as the previous experiments, for every dataset, they are two classes, 5000 instances for each, each instance have 100 features (25 dependent features and 75 independent features) and every feature has three possible values (Value1, Value2 and Value3). All features have the same distribution for both classes. The only difference between two subsets is the conditional probability distribution of the feature 98. This means that only Feature98 and the features it depends on can discriminate the two classes. Feature Feature98 and the features it depends on are defined as the relevant feature and this is why our wrapper feature selection approach will try to select them and only them for classification. Table 6 is the summary of dataset o, p and q.

Dataset	O	P	q
Class	2	2	2
Instance	5000x2	5000x2	5000x2
Feature Number	100	100	100
Dependent feature	25	25	25

Number			
Independent feature Number	75	75	75
Relevant feature Number	3	4	5
Dependencies among relevant feature	2	3	4
Relevant feature	Feature1, Feature8, Feature98	Feature1, Feature8, Feature9, Feature98	Feature1, Feature8, Feature9, Feature11, Feature98

Table 6. The summary of dataset o, p and q

As an example, Fig. 14 and Fig. 15 are the pair of Bayesian Networks which we have used to generated the test dataset o, one network for class0 and one for class1. Each Bayesian Network is the probabilistic model used to represent the joint distribution of one class dataset. The nodes of the Bayesian Network stand for the features of the datasets. Node1 is corresponding to Feature1 in the dataset; Node2 is corresponding to Feature2, and so on. The dependency properties are coded as a direct acyclic graph (DAG) in the Bayesian Network and the arcs correspond to direct influences between the features. The conditional probability table (CPT) of the nodes are used to describe the conditional probability distribution of the features. For dataset p and q, the only difference is that Feature98 also depend on Feature9 for dataset p and on Feature9 and Feature11 for dataset q.

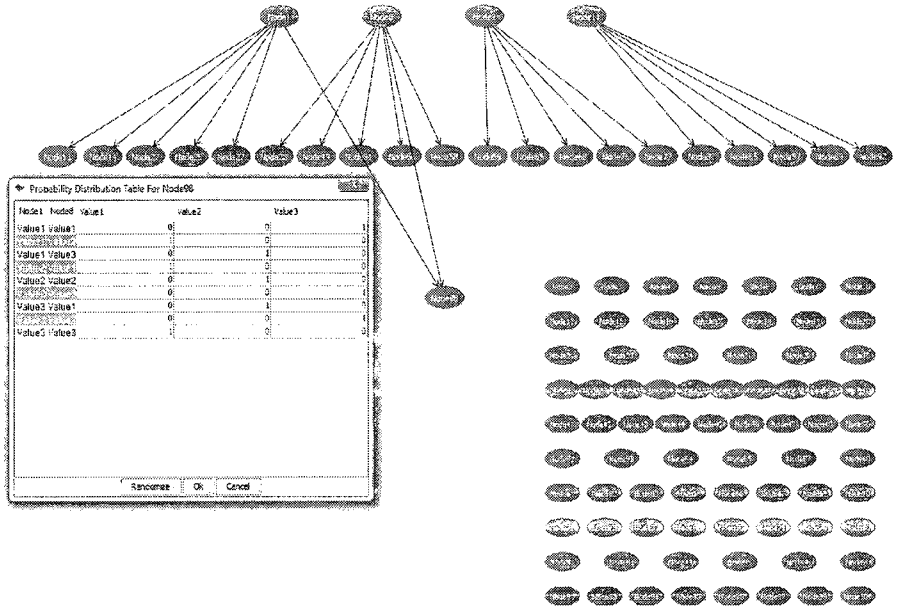

Fig. 14. Bayesian network structure for relevant dependent features of class 0 of dataset o

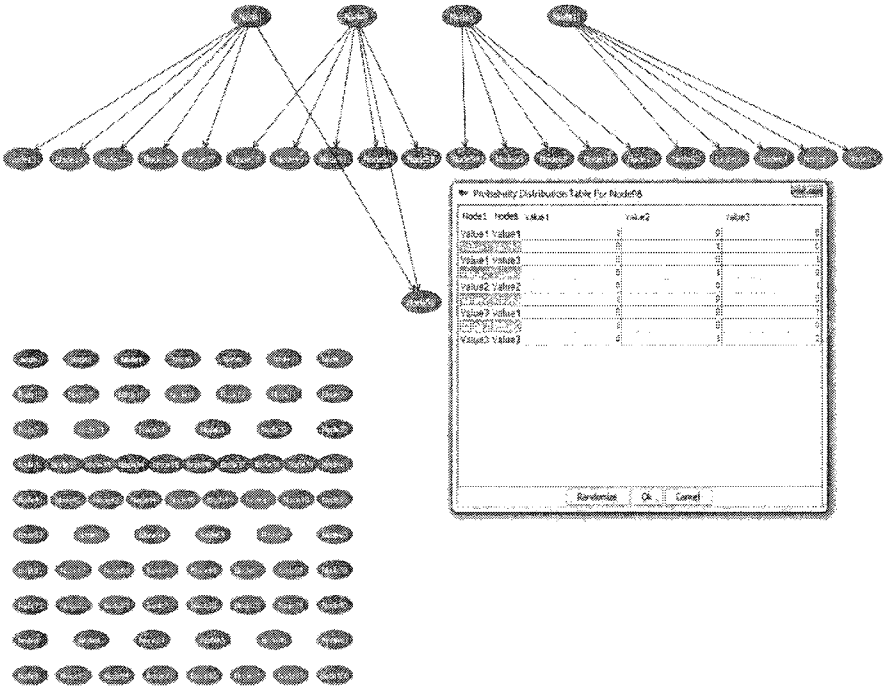

Fig. 15. Bayesian network structure for relevant dependent features of class 1 of dataset o

Experiments

Because the SVM classifier is time consuming, the experiments of this and next Chapter are on the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca) by using parallel computation. The capability of different feature subset selection approaches to extract the relevant features for varying dependencies among the relevant features have been tested based on dataset o, p and q. In these experiments, for the filter model, the feature selection approaches used are Relief and CFS. Again, the threshold τ of Relief is set to 0, which mean all the features which value have any connection with the target class will be selected, even though the connection is a tiny one. The programs are implemented by Weka [27]. For the Wrapper model, Naïve Bayes is one of the chosen classifier. It is combined with GAs or BOA search algorithm respectively (NB-GA, NB-BOA). SVM is the other chosen classifier which is implemented by Chang et al. [30]. The kernel function SVM is RBF. BOA is coded by Pelikan [28]. These feature selection approaches are SVM-BOA, SVM-GA. The last feature selection approach is SVM-mBOA which combines SVM-BOA and multi-objective optimization method together. The fitness function of NB-BOA and NB-GA is same as the previous one:

$$fitness = Acc./n^p$$

For these experiments the adjustment parameter p is set to 0.01(the biggest of the previous experiments in Chapter III, to maximally constrain selected features). The goal is to find small feature set with high classification accuracy. But how to set p is subjective process which should based on lots of experiments and knowledge. To avoid guessing a p value is one problem that SVM-mBOA intended to solve in our experiments.

Datasets	relief	CFS	SVM-BOA	SVM-mBOA	SVM-GA	NB-BOA	NB-GA
o	3/69	0/28	3/25	3/3.7	3 /4.5	0/18	0/1
p	4/76	1/32	3.7/19	2.8/3.8	4/4	0/18	0/1
q	1/76	1/32	0/11	0.7/3.7	3/6	0/16	0/1

Table 7. Experiments based on datasets o,p and q

Datasets	relief	CFS	SVM-BOA	SVM-mBOA	SVM-GA	NB-BOA	NB-GA
o	0.5091	0.4997	0.8217	0.9999	1.000	0.5006	0.5113
p	0.5114	0.5002	0.8381	0.8415	0.9998	0.5038	0.5120
q	0.5083	0.5009	0.5247	0.5245	0.8438	0.5019	0.5113

Table 8. Classification accuracy of the experimens based on datasets o, p and q

As we motioned before, For dataset o, p and q, all features have same distribution for both class. The only difference between two subsets is the conditional probability distribution of feature Feature98. Classifier should use the relevant features (feature98 and its depended features) to discriminate the two classes. Dataset o has 3 relevant features, p has 4, and q has 5. Table 7 shows the results of the feature selection approaches to extract the relevant features. Same as before, 3/3 means "3 relevant features selected among / 3 selected features".

Table 8 shows the results of classification accuracy by using the selected features of Table 7. The classifier is SVM with 2 fold cross validation. For those feature selection approaches using BOA or GA, the value in the Table 7 and

Table 8 is an average value over 3 times runs. We can see that:

- Comparing the performance of these features selection approaches, SVM-GA is the best one based on almost all datasets except o. It can find most of the possible relevent features with a almost minimal set of selected features to get the highest classification accuracy. SVM-mBOA is also quite good, second after SVM-GA. In the experiments based on dataset o and p, Relief finds some relevant features also but with much more false positive, that is it selects much more irrelevant features (experiments based on data set o and p, the results are 3/69 and 4/76). Using these features selected by relief, the classification accuracy is just around 0.5, this is not very meaningful for classification. So, for these datasets in which there are complex dependencies among the relevant features and these features have similar distribution for both classes at same time, Relief can not handle it whereas the wrapper approaches based on BOA can.
- For SVM-BOA, SVM-mBOA and SVM-GA, we can see that higher the dependency level among the relevant features is, the more complex the task of finding the relevant features will be. For example, SMV-mBOA can find average 3 relevant features over 3.7 selected features based on dataset o. However, it can only find average 0.7 relevant features over 3.7 selected features on dataset p. The classificaion accuracy by using the selected features also decreases form 0.9999 to 0.5247.
- In our experiments, NB-BOA and NB-GA are unable to find the relevant features (0/18 or 0/1), the classification accuracies are also around 0.5, this mean that they are not sutiable for these datasets. It can be explained be the fact that Naïve bayes classifier is not able to represent the dependencies between features, it considers

every feature as independent, therefore, it is not able to use these relevant features to discriminate classes. The performance of CFS is poor in these situations as well.

- According to the experiments based on SVM-mBOA and SVM-GA on dataset o and p, we see that the wrapper model feature selection prefer to select the relevant features (e.g. feature1,feature8 and feature98 in dataset o) . For instance, experiments based on dataset o, SVM-mBOA averagely selected 3.7 features in which 3 features are relevant. Using these selected features, the classifcaion accuracy is 0.9999. The other features which are useless to classification are ignored.

- SVM-GA has a better performance than SVM-BOA on datasets o, p and q. The results are (3/4.5,1.000) vs (3/25,0.8217), (4/4,0.9998) vs (3.7/19. 0.8381) and (3/6,0.8348) vs (0/11,0.5247). Here, the "(m/n, a)" means "(m relevant features selected among / n selected features, classifcaion accuracy a)". The reasons may be the diversity of population of BOA decreases faster than GA's (GA has a mutation operator, whereas, BOA does not). To solve this problem, one solution is to add a mutation operator to the BOA. But it will impair the key substructures or building-blocks of the optimal solutions and this will give up the one important benefit of BOA. In our experiments, we combine a Multi-objective optimization method into SVM-BOA which not only avoid setting the value of the punishment parameter p in the fitness function, it can also help us increasing the diversity of population in every generation. When using this method, the classification accuracy is not the only criterion to evaluate two different feature sets (solutions). For example, right now we cannot say that one solution which has fewer classification accuracy but less features

is worse than a solution which has higher classification accuracy with more features. This mean that these solutions which are in the Pareto front and have fewer classification accuracy but less features will not be automatically discarded as previously did. They will be kept in the next generation. This can help keeping the population diversity and does not lose any useful information and break the BBs. We can also remark that for SVM-mBOA on datasets o, p and q compared to SVM-BOA, the results are (3/3.7,0.9999) vs (3/25,0.8217), (2.8/3.8,0.9998) vs (3.7/19. 0.8381) and (0.7/3.7, 0.8348) vs (0/11,0.5247). It shows that the Multi-objective optimization method can effectivelly improve the search ability of BOA in our experiments. The results obtained with SVM-mBOA are still not as good as the ones obtained with SVM-GA. It is still not clear why it is the case at this phenomenon should be investigate in further works.

CHAPTER V

EXPERIMENTS BASED ON VARIED DEPENDENCY AMONG THE IRRELEVANT FEATURES

Datasets

In this section, the experiments are based on dataset r, s and t, which have diffreent dependencies among the irrelevent features. They are also generated by three pairs of Bayesian Networks. This is similar as the previous experiment (chapter IV). The difference is that the dataset r only have 25 features. All of them are mutually dependent. The dataset s has the same 25 dependent features as the ones in dataset r but also another 75 independent features. The dataset t is similar to dataset s but the the other 75 features are mutually dependent. **Fig. 16**, **Fig. 17** and **Fig. 18** are the illustration of dataset r, s and t. Feature1, feature8, feature9 and feature98 are the relevant features for all datasets r,s and t. As in the previous chapter, for every dataset, all features have same distribution for both class. The only difference between two subsets is the conditional probability distribution of feature98. **Table 9** is the summary.

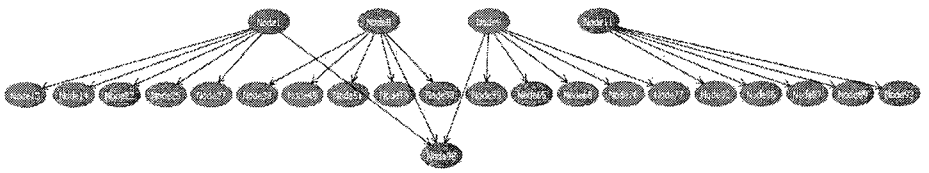

Fig. 16. The Bayesian Network example for Dataset r

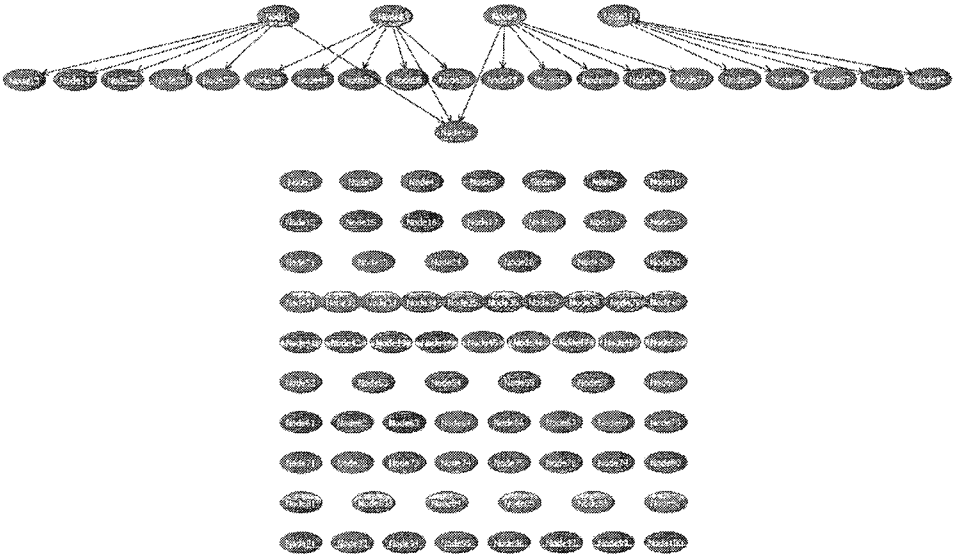

Fig. 17. The Bayesian Network example for Dataset s

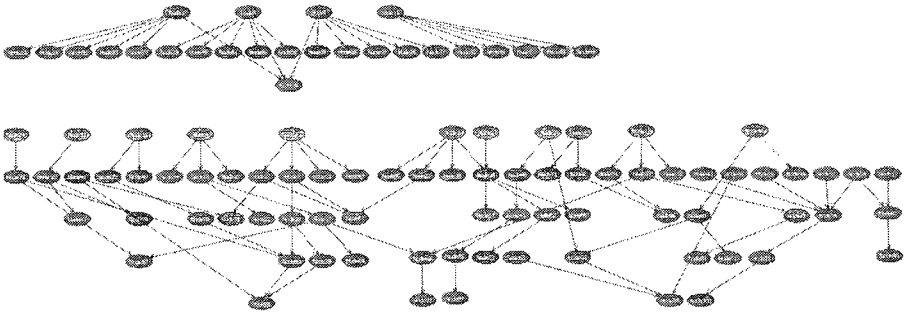

Fig. 18. The Bayesian Network example for Dataset t

Dataset	r	s	t
Class	2	2	2
Instance	5000x2	5000x2	5000x2
Feature Number	25	100	100

Dependent feature Number	25	25	100
Independent feature Number	0	75	0
Relevant feature Number	4	4	4
Dependencies among relevant feature	3	3	3
Relevant feature	Feature1, Feature8, Feature9, Feature98	Feature1, Feature8, Feature9, Feature98	Feature1, Feature8, Feature9, Feature98

Table 9. The summary of dataset r, s and t

Experiments

The goal of this experiment is to find how changing dependencies among the irrelevant features can affect the performance of relief, SVM-BA, SVM-mBOA, SMV-BOA to extract the releant features. The experiments are based on dataset r, s and t.

Datasets	relief	CFS	SVM-BOA	SVM-mBOA	SVM-GA	NB-BOA	NB-GA
r	4/4	1/1	4/4	4/4.3	4/4	0/1	0/1
s	4/76	1/32	3.7/19	2.8/3.8	4/4	0/18	0/1
t	1/1	1/1	3.3/52	3.3/6	4/6	0/1	0/1

Table 10 Experiments based on datasets r, s and t

Datasets	relief	CFS	SVM-BOA	SVM-mBOA	SVM-GA	NB-BOA	NB-GA
R	0.9998	0.4946	0.9998	0.9998	0.9998	0.4906	0.4905
s	0.5114	0.5002	0.8381	0.8415	0.9998	0.5037	0.5012
t	0.4946	0.4946	0.5962	0.9045	0.9990	0.4924	0.4919

Table 11 Classification accuracy of the experimens based on datasets r, s and t

From
Table 10 and
Table 11 we can see that:

- Similarly to the experiments of previous chapter, SVM-GA is the best one to find the relevant features with highest classification accuracy in these experiments based on dataset r, s and t. SVM-mBOA and SVM-BOA are the second and third one. The difference is when the dataset only include 25 features (dataset r), Relief works pretty well for finding relevant features and discriminating two class very precisely (4/4, 0.9998). Here, the "(m/n, a)" means "(m relevant features selected among / n selected features, classification accuracy a)". However, if the number of features is increased, or the irrelevant features are mutually dependent, that should affect the capability of Relief to extract the relevant features for classification. This is because Relief also considers some kind of dependency in some degree when it calculates each solution's Near-hit instance (the closest instance according to Euclidean Distance in same class), and Near-miss instance (the closest instance according to Euclidean Distance in

opposite class). If there are lots of dependencies among irrelevant features, it will also change the distance's value and affect the performance of Relief.

- The experiments show that there are some relation between the variing dependencies among the irrelevant features and the results of SVM-BOA and SVM-mBOA to extract the relevant features. For dataset s, and t, they have identical 25 dependent features which include 4 relevant features; but the distributions of the other 75 features are different for each dataset. Using SVM-BOA or SVM-mBOA, the results they got are different based on different dataset s and t, say (SVM-BOA: 3.7/19, 0.8381 vs 3.3/52, 0.5962) or (SVM-mBOA: 2.8/3.8, 0.8415 vs 3.3/6, 0.9045), However, there is not much difference of performance for SVM-GA for the same situations (4/4,0.9998 vs 4/6,0.9990). It means that those dependencies can affect BOA to generate the probabilistic model and then alter the experiment outcome of SVM-BOA and SVM-mBOA. On the whole, SVM-GA and SVM-mBOA got pretty good results on all dataset r , s and t. Especially for dataset t (4/6, 0.9990 and 3.3/6 0.9045).

CHAPTER VI

CONCLUSION AND FUTURE WORKS

First of all, we presented a novel method which used the pre-designed Bayesian networks to generate candidate datasets to simulate different classification situations to test feature selection approaches. Compared with the real or the artificial datasets which are often used for feature selection evaluation, our method is simpler and more accurate. The different datasets can be easily generated only by changing some parameters of the Bayesian Networks.

Second, according to our experiment, we found that more dependencies or more complex network of dependencies among the relevant features will greatly affect the capability to find the relevant features for classification. The higher dependency level, the more complex of the task is.

Third, the multi-objective optimization method not only helps to choose a trade-off solution from conflict objectives, but it also can help keeping the diversity of the populations in each generation and improve the overall quality of solutions for BOA in our experiments.

Finally, Relief usually is a very effective and efficient feature selection method. This has been proven in our experiments in which that the datasets which have been generated by partially random Bayesian Network. Relief got the best result. One well known drawback of Relief is that it is difficult to filter redundant features. However, we found another limitations of Relief that it cannot handle sophistic dependencies among the relevant features and these features have similar distribution for both classes at same

time; moreover, the number or the dependencies among the irrelevant feature can also affect the capability of Relief.

In our experiments, there are still some open questions that have not been answered. For example, we do not know why GA do better than BOA and why most unselected features are represented by leaf or closed to leaf nodes of Bayesian network. It should be interesting research and experiments in future works.

REFERENCES

1. Yang, Y. and J.O. Pedersen, *A comparative study on feature selection in text categorization*, in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*. 1997, Citeseer. p. 412-420.
2. Xing, E.P., M.I. Jordan, and R.M. Karp, *Feature selection for high-dimensional genomic microarray data*, in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*. 2001, Citeseer. p. 601-608.
3. Saeys, Y., I. Inza, and P. Larrañaga, *A review of feature selection techniques in bioinformatics.* Bioinformatics, 2007. **23**(19): p. 2507.
4. Gras, R., *How efficient are genetic algorithms to solve high epistasis deceptive problems?*, in *IEEE Congress on Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence)*. 2008. p. 242-249.
5. Asuncion, A. and D.J. Newman. *UCI Machine Learning Repository*. 2007.
6. Thrun, S.B., et al. *The MONK's Problems A Performance Comparison of Different Learning Algorithms*. 1991.
7. Heckerman, D., D. Geiger, and D.M. Chickering, *Learning Bayesian networks: The combination of knowledge and statistical data.* Machine learning, 1995. **20**(3): p. 197-243.

8. Cheng, J., et al., *Learning Bayesian networks from data: an information-theory based approach.* Artificial Intelligence, 2002. **137**(1-2): p. 43-90.

9. Salehi, E., J. Nyayachavadi, and R. Gras, *A Statistical Implicative Analysis Based Algorithm and MMPC Algorithm for Detecting Multiple Dependencies*, in *The 4th Workshop on Feature Selection in Data Mining (FSDM)*. 2010. p. (JMLR-WCP) 10:22-34.

10. Saheli E. and Gras R., *An empirical comparison of the efficiency of several local search heuristics algorithms for Bayesian network structure learning*, in *Learning and Intelligent OptimizatioN IEEE international conference*. 2009.

11. Dash, M. and H. Liu, *Feature selection for classification.* Intelligent data analysis, 1997. **1**(3): p. 131–156.

12. Kohavi, R. and G.H. John, *Wrappers for feature subset selection.* Artificial Intelligence, 1997. **97**(1-2): p. 273–324.

13. Dash, M. and H. Liu, *Consistency-based search in feature selection.* Artificial Intelligence, 2003. **151**(1): p. 155–176.

14. Hall, M.A., *Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning*, in *ICML '00: Proceedings of the Seventeenth International*

Conference on Machine Learning. 2000, Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA. p. 359-366.

15. Kira, K. and L.A. Rendell, *The feature selection problem: Traditional methods and a new algorithm*, in *Proceedings of the National Conference on Artificial Intelligence*. 1992. p. 129-129.
16. Quinlan, J.R., *Induction of decision trees.* Machine learning, 1986. **1**: p. 81–106.
17. Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern classification*. 2001: Citeseer.
18. McCallum, A. and K. Nigam, *A comparison of event models for naive bayes text classification*, in *AAAI-98 workshop on learning for text categorization*. 1998, Citeseer.
19. Korf, R.E., *Depth-first iterative-deepening.* Artificial Intelligence, 1987. **27**(1): p. 97–109.
20. Narendra, P.M. and K. Fukunaga, *A branch and bound algorithm for feature subset selection.* IEEE Transactions on Computers, 1977. **100**(26): p. 917–922.
21. Mitchell, M., *An introduction to genetic algorithms.* 1998.
22. Aarts, E. and J. Korst, *Simulated annealing and Boltzmann machines*. 1989: John Wiley & Sons New York.

23. Inza, I., et al., *Feature subset selection by Bayesian network-based optimization.* Artificial Intelligence, 2000. **123**(1-2): p. 157–184.

24. Lima, C.F., et al., *Influence of selection and replacement strategies on linkage learning in BOA*, in *Proceedings of the 2007 Congress on Evolutionary Computation CEC*. 2007. p. 1083–1090.

25. Pelikan, M., D.E. Goldberg, and E. Cantu-Paz, *BOA: The Bayesian optimization algorithm*, in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*. 1999. p. 525–532.

26. Press, W.H., et al., *Numerical recipes in C*. 1992: Cambridge Univ. Press Cambridge MA, USA:.

27. Hall, M., et al., *The WEKA Data Mining Software: An Update.* SIGKDD Explorations, 2009. **11**.

28. Pelikan, M., *A simple implementation of the Bayesian optimization algorithm (BOA) in C++(version 1.0).* IlliGAL Report, 1999. **99011**.

29. Ebert-Uphoff, I., *Measuring Connection Strengths and Link Strengths in Discrete Bayesian Networks.* Woodruff School of Mechanical Engineering, Antlanta, GA, Tech. Rep, 2006.

30. Chang, C.C. and C.J. Lin, *LIBSVM: a library for support vector machines*. 2001: Citeseer.

VITA AUCTORIS

NAME:	Qin Yang
PLACE OF BIRTH:	ShiChuan, CHINA
YEAR OF BIRTH:	1971
Education:	East China Shipbuilding Industrial Institute
	1989-1993
	University of Windsor, Windsor, Ontario
	1999-2001 M.Sc.